title: Understanding Bundles

import Tabs from '@theme/Tabs' import TabItem from '@theme/TabItem' import SendBundleRpc from '@site/docs/specs/mev-share/_mev_sendBundle.mdx' import BuilderInheritance from '@site/docs/specs/mev-share/blurbs/_builderInheritance.mdx' import Hints from '@site/docs/specs/mev-share/HintsTable'

Bundles on MEV-Share are conceptually the same as bundles on MEV-Boost: they are an ordered array of transactions that execute atomically. However, their structure is a bit different. MEV-Share bundles use a new method called mev_sendBundle which has additional fields used to specify privacy preferences and introduce other new features like post-execution validity checks.

Bundle Definition

MEV-Share Bundles have the following structure:

Key Fields: * **inclusion**: Defines the pre-inclusion predicates to check (e.g. block range). ***body**: Contains transactions, bundle hashes, or transaction hashes. * **validity**: (Optional) Defines the post-inclusion predicates to check, which are just refund parameters at the moment. * **privacy**: (Optional) Sets privacy configurations, including which builders to submit to. NOTE: the Flashbots builder is submitted to by default even if only other builders are specified. Optional properties are denoted with a ?.

:::info Note to searchers on builders

:::

This is the generic bundle structure used in MEV-Share. This comprehensive specification enables several exciting features, outlined in the next section.

Sharing hints

MEV-Share bundles can share hints with searchers, which can be used to backrun the bundle. This is done by setting the privacy parameter in mev_sendBundle. The privacy parameter is an object with the following fields:

Searchers can share hints to give other searchers information about their bundle that would allow them to be backrun. If your bundle gets backrun by another searcher, you get paid a cut of the MEV they extract!

Builders

MEV-Share bundles can be sent to multiple builders at once. This is done by setting the builders field in the privacy parameter of mev_sendBundle.

Bundle composition (backrunning other bundles)

With the privacy parameter in mev_sendBundle you can share select information about your bundle with other searchers, who can then use that to try to extract MEV. Should they succeed, you get paid some of the MEV they extracted!

One example that works well with bundle composition is a liquidation bot. Liquidations often cause a price shift, leaving MEV on the table which can be captured by arbitrage with a backrun bundle. If you run a liquidation bot, you can earn more MEV by sending your bundles to MEV-Share with the tx_hash hint enabled, which will allow other searchers to backrun your bundle.

An example would look something like this:

typescript const params: BundleParams = { inclusion: { block: 17539448, maxBlock: 17539458 }, body: [{tx: "0x02...", canRevert: false}, {tx: "0x02...", canRevert: false},], privacy: { hints: { txHash: true, }, }, }

Specifying the tx_hash hint in your bundle shares the hashes of your bundle's transactions with searchers on MEV-Share, which is what allows them to backrun your bundle. Again, when they do this, you earn a cut of the profit!

You may also try experimenting with other hints to give searchers more data with which to formulate a backrun. Sharing more data will lower your privacy, but will make your bundle easier to backrun, and increase the likelihood of your bundles earning extra MEV.

:::caution only original transactions are supported

Bundles that set the privacy parameter can only contain original signed transactions in the body parameter. Bundles using transactions specified by {hash} are not allowed to use the privacy parameter (full bundle privacy is maintained in this case).

Allowing such bundles to share data using the privacy parameter would compromise the privacy guarantees of user transactions.

:::

See Sending Bundles for more information.

Now that we know all the different ways in which we can send and share bundles, we're finally ready to send a bundle.